

AI Generated Face Detection Using DenseNet - 121

Sandhya Adhikari*, Rujit Shrestha, Kalyan Subedi, Roshna Shrestha, Sushant Pandey

Himalaya College of Engineering, Tribhuvan University, Nepal

*Corresponding Author: hce078bct038@hcoe.edu.np

Abstract

In this digital era, AI-generated faces have become so realistic that distinguishing real from fake has become almost impossible. To address this issue, the findings presented in this paper were utilized to develop an advanced digital detection mechanism. The paper presents a DenseNet-121-based neural network trained on 65,000 real and synthetic facial images. The paper implemented the detection into a practical, user-friendly web application based on Flask and React, where users can upload their images, do necessary cropping, and instantly receive predictions along with confidence scores, essentially democratizing deepfake detection to the masses. When evaluating on 13,000 entirely new sets of images, it performed 76 errors (65 real faces wrongly flagged as fake, 11 fakes wrongly accepted), attaining 99.42% accuracy in just seven training epochs by using techniques such as data augmentation and check pointing, including image augmentation and adaptive check pointing. These results confirm the model's effectiveness in distinguishing between genuine and AI-generated faces. The paper then concludes with future directions, including other video deepfake detection and ethical considerations concerning this technology, asserting that without enhanced digital literacy and public awareness programs, building safer and more trustworthy digital worlds would only be possible by having effective detection tools.

Keywords: Deepfake Detection, Synthetic Face, Classification, DenseNet-121, CNN, Face Verification, AI-Generated Images

1. Introduction

In the modern digital landscape, photos and videos are ubiquitous across platforms such as social media, websites, and various online spaces. It offers imagination the essential room, yet also generates significant distress. The rise of advanced AI technologies, such as Generative Adversarial Networks (GANs), DALL-E, and Midjourney, has made the creation of strikingly realistic synthetic faces fairly easy nowadays. The faces created are often realistic and believable to such a degree that it becomes very difficult, if not impossible, to tell them apart from real human faces. These occurrences have serious implications since they lead to the spread of false information, facilitate identity theft, and impair trust in on-line content. The accuracy and precision of current face-detection models are significantly challenged.

Additionally, many of the models in use today lack transparency because they typically do not provide a rationale for their classification outcomes. This lack of open-ness is particularly troubling in critical application areas where even small mistakes can have catastrophic results, such as identity protection, security, and media verification. This paper primarily aims to address this urgent problem. This report uses a convolutional Neural Network (CNN) method incorporating DenseNet-121 to accurately determine whether a specific face belongs to a genuine human or a synthetic one.

Additionally, the model provides confidence levels in addition to binary classification, thus assessing its certainty in both categories. The main goal is to develop strong media manipulation detection abilities, alongside skills that collectively support the wider objective of fostering a safer and more trustworthy digital environment.

2. Literature Review

Face detection has advanced rapidly due to its critical role in applications such as security, biometric identification, and content moderation. Nevertheless, the introduction of Generative Adversarial Networks (GANs) and diffusion models dampened the difference between real and synthetic faces, causing legitimate concerns over digital media authenticity, according to [1].

Traditional face detection methods, including Haar cascades and template matching opened the doors to real-time face detection at first but were slowed down by differences in environment such as variation in light, head pose variation and partial face blocking. The Viola-Jones algorithm [2], based on Haar-like features and cascade classifiers, was a milestone but it broke down under real-world scenarios. Statistical approaches like Eigenfaces and Fisherfaces, based on principle component analysis (PCA) and linear discriminant analysis (LDA) were limited to controlled environments.

The limitation of such methods caused the emergence of deep learning, particularly Convolutional Neural Networks (CNNs) that learn patterns and extract features automatically from large datasets. CNN models such as AlexNet, VGG, and ResNet significantly improved detection precision [3] while real-time detectors like Faster R-CNN and YOLO pushed performance even higher by applying region-based proposals and optimizing speed according to [4].

This work utilizes DenseNet-121, a newer CNN architecture that includes dense connections between layers to promote gradient transfer and reuse of features [3]. DenseNet-121 is parameter efficient over ResNet and provides more accuracy, especially where data is limited. Adam optimizer was utilized because of its adaptive learning rate and momentum updates

that enable faster and more stable convergence during training as stated in [5]. Recent advances attempt to use architectures such as MobileNet to leverage attention mechanisms for improved feature detection and fuse multi-modal signals (eg., audio-video) for robust detection. Most of the present systems are however, plagued by weak training data and interpretability issues when confronted with high-quality synthetic faces. It has also been studied that face analysis systems are biased towards gender as well as race, and there exists a need for fairness in detection mechanisms [6]. In related tasks like cartoon face recognition, deep residual networks have also shown improved performance over traditional CNNs, further supporting the potency of architectures like ResNet and DenseNet [7]. Similarly, public datasets like WIDER FACE [8] and Fake Face in the Wild (FFW) [9] have been central in testing and validating face detection systems generally in synthetic face recognition. However, these methods often under perform under diverse lighting conditions, which our method addresses.

This paper addresses these challenges with a DenseNet-121-centered detection frame-work, Adam-Optimized, and supported by a web-based React-Flask interface for convenient real time operation. At 99.42% validation accuracy, the system demonstrates great potential for practical and ethical applications in AI-generated face detection.

Recent advancements in deepfake detection have explored transformer-based architectures such as Vision Transformers (ViT) and hybrid CNN-transformer models, which demonstrate improved global feature extraction and robustness against high-quality synthetic images. Additionally, diffusion model-based image generation has introduced new challenges, leading to the development of specialized detection techniques targeting diffusion artifacts. These modern approaches highlight the evolving landscape of AI-generated image detection and provide context for the relevance of the proposed DenseNet-121-based framework.

3. Methodology

The methodology follows a structured pipeline consisting of five major stages: dataset collection, data pre-processing, model architecture and training, evaluation, and web application development. This pipeline ensures systematic development, optimization, and deployment of the AI-generated face detection system.

3.1 Dataset Collection

A total of 65,000 facial images were collected from publicly available datasets to ensure a balanced representation of real and synthetic faces. Real images were obtained from the Flickr-Faces-HQ (FFHQ) dataset [10], while synthetic images were sourced from the Generated Faces in the Wild (GFW) dataset [11].

The dataset was equally divided into 32,500 real and 32,500 AI-generated images. To ensure robust model training and evaluation, the dataset was partitioned into:

- 1) Training set: 80% of the data
- 2) Validation set: 20% of the data

All images were stored locally to enable efficient data loading and pre-processing.



Figure 1. Example of Real Face (FFHQ) Used in Data Baset



Figure 2. Example of AI-Generated Face Used in Data Baset

The selected datasets (FFHQ and GFW) are widely used and publicly available benchmarks in face generation and detection research, ensuring data authenticity, diversity, and reliability for training and evaluation.

3.2 Training Strategy and Check pointing

Training was done locally on a system consist-ing of an Intel i5 11th gen CPU, RTX 3050 GPU and 16 GB of RAM. After every epoch, performance metrics and the model state was saved, capturing the model weights and biases, optimizer and scheduler states, current loss, best validation accuracy, and the epoch index, ensuring that training could be resumed seamlessly.

3.3 Data Pre-processing

Image pre-processing was performed using Py-Torch. All images were resized to 224×224 pixels to match the input dimensions expected by the DenseNet-121 architecture pretrained on ImageNet.

To improve the model's robustness and reduce overfitting, several data augmentation techniques were applied during training. Random horizontal and vertical flips, along with random rotations of up to $\pm 10^\circ$, were introduced to en-sure the model remained invariant to orientation changes that naturally occur in real-world facial images.

A random crop with 4-pixel padding was applied to encourage the model to focus on varying regions of the face rather than relying on fixed spatial positions. Color jitter adjustments to brightness, contrast, saturation, and hue were incorporated to simulate varying lighting and camera conditions, improving generalization across diverse image sources.

Gaussian blur with a kernel size of 5 and $\sigma \in [0.1, 2.0]$ was applied to replicate slight focus variations and compression artifacts commonly found in real-world images.

Finally, all images were converted to tensors and normalized using the ImageNet mean and standard deviation, ensuring that the input distribution aligned with the pretrained model's expected range and facilitating stable and efficient gradient updates during fine-tuning.

3.4 Model Architecture and Training

DenseNet-121 is a convolutional neural net-work architecture characterized by dense connectivity between layers, where each layer receives feature maps from all preceding layers. This design promotes feature reuse, improves gradient flow, and reduces the vanishing gradient problem. The architecture consists of multiple dense blocks interconnected by transition layers that perform convolution and pooling operations. In this work, a pretrained DenseNet-121 model was adopted and fine-tuned for bi-nary classification of real and AI-generated faces.

A DenseNet-121 pretrained on ImageNet served as the base model, with all layers frozen except the final five convolutional blocks and the classifier. The unfrozen layers were fine-tuned using the Adam optimizer with a learning rate of 10^{-3} and a weight decay of 10^{-4} .

Cross-entropy loss was used as the objective function, and a ReduceLROnPlateau scheduler was employed to adaptively

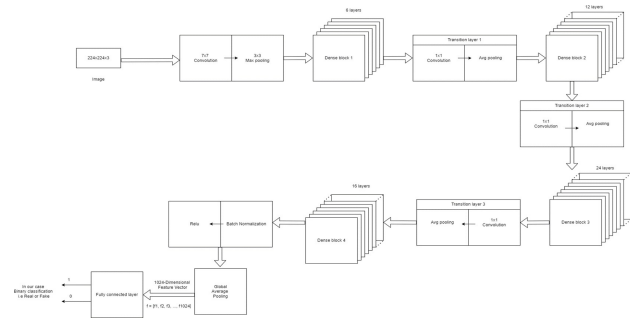


Figure 3. DenseNet-121 Architecture Showing Dense Blocks and Transition Layers. Each Layer Receives Feature Maps From All Preceding Layers, Enabling Feature Reuse and Efficient Gradient Flow

reduce the learn-ing rate by a defined factor whenever the validation loss plateaued over a specified number of epochs, preventing the optimizer from overshooting minima during the later stages of training.

To improve computational efficiency on the available hardware, mixed-precision training was utilized via GradScaler and autocast. The autocast context manager automatically cast operations to 16-bit floating point where appropriate, reducing memory consumption and accelerating computation, while GradScaler dynamically scaled the loss during backpropagation to prevent gradient under-flow that can arise when operating in reduced precision. Together, these techniques ensured training stability while making effective use of the constrained hardware resources available. Training and validation used batch sizes of 32 and 64, respectively. Each epoch followed these steps:

- 1) Forward pass
- 2) Loss computation
- 3) Backward pass
- 4) Parameter update
- 5) Gradient-free validation

3.5 Evaluation

A set of 13,000 images (6,500 real and 6,500 synthetic), representing 20% of the dataset, was used to assess model performance.

Performance metrics, including a confusion matrix and other evaluation measures, were generated using a dedicated evaluation script.

3.6 Web Interface Design and Integration

The front end was implemented in React, featuring drag-and-drop uploads, a cropping inter-face, loading animations, and a responsive lay-out built using structured grid and flexible box-based styling. State management was handled through built-in framework hooks, while asynchronous form-based data transmission was used to send images to the backend. The backend was developed using a lightweight framework, applying the same pre-processing pipeline used during training, performing inference using the trained model weights, and returning structured responses containing the predictions and confidence scores.

4. Results and Analysis

The development and evaluation of the AI-generated face detection system utilizing the DenseNet-121 architecture has yielded promising results. The model was trained on a balanced dataset comprising 65,000 images (32,500 real and 32,500 fake), with an 80–20 split for training and validation, respectively. Training was conducted over 7 epochs, achieving a peak validation accuracy of 99.42% in the fifth epoch, with a corresponding training accuracy of 98.44%. This performance is summarized in Figures 3, 4, and 5, which highlight the progressive reduction in loss and improvement in accuracy across both the training and validation phases.

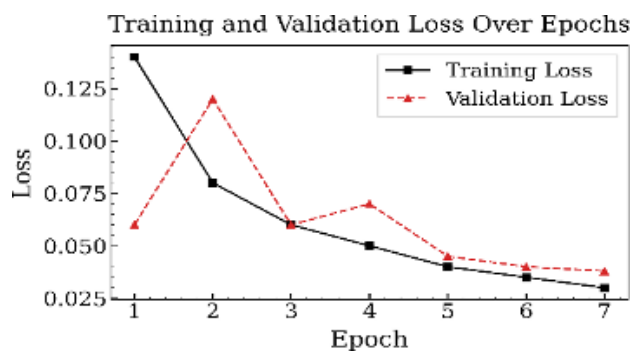


Figure 4. Performance Graph of Training and Validation Loss

As shown in Fig. 4, the training loss decreased steadily from 0.14 at epoch 1 to approximately 0.03 at epoch 7 demonstrating consistent model convergence throughout training. The validation loss exhibited a transient spike at epoch 2, rising to 0.12, before declining and stabilizing near 0.038 by epoch 7. This initial instability is attributed to the model's early difficulty in generalizing to unseen data, which resolved as the training progressed. Beyond epoch 5, both loss curves converged with marginal reductions every epoch, indicating that the model had reached near optimal state. Training was therefore terminated at epoch 7, as further iterations were unlikely to yield significant performance gains and posed an increased risk of overfitting.

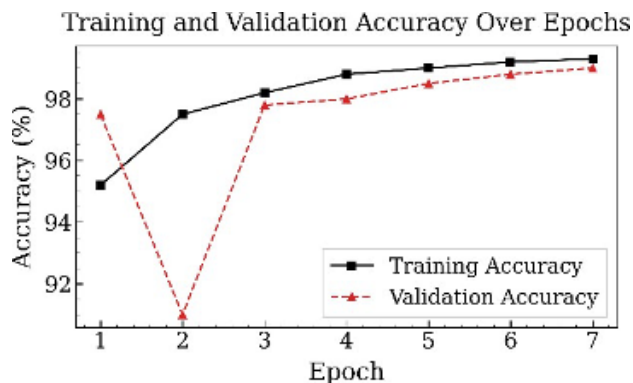


Figure 5. Performance Graph of Training and Validation Accuracy

Fig. 5 further illustrates that the training accuracy increased monotonically from 95.2% at epoch 1 to 99.3% at epoch 7, reflecting the model's progressive improvement in classifying training samples. The validation accuracy showed a sharp

decline to 91.0% at epoch 2, consistent with the loss spike observed during the same epoch, before recovering and reaching 99.0% by epoch 7. From epoch 3 onward, both accuracy curves converged steadily, with a final gap of only 0.3% between training and validation accuracy. This close agreement indicates strong generalization performance and minimal overfitting on the classification task.

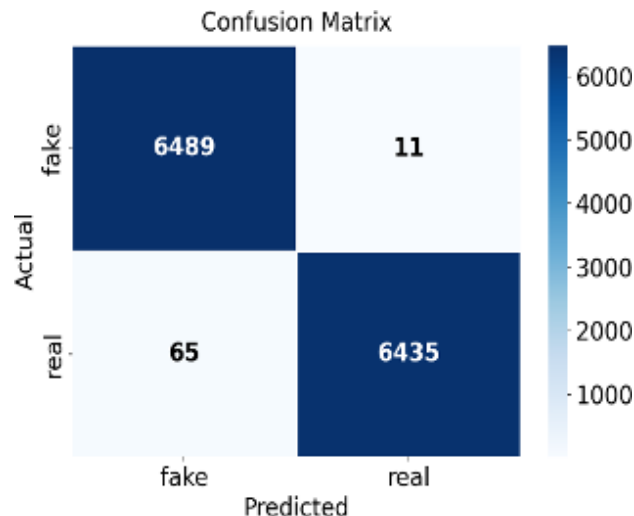


Figure 6. Confusion Matrix From Evaluation

The confusion matrix in Fig. 6 summarizes the classification results of the model. Out of the total instances, 6,435 were correctly classified as real (true positives) while 6,489 were classified as fake (true negatives). The model incorrectly classified 11 fake instances as real (false positives) and 65 real instances as fake (false negatives). These results demonstrate a high level of accuracy in distinguishing between real and fake classes with a very low misclassifications rate.

Table 1. Evaluation Metrics

Metric	Value
Precision	0.9983
Recall	0.9900
F1 Score	0.9941
Accuracy	0.9949

The high values of true positives and true negatives, combined with a minimal occurrence of misclassifications, illustrate the strong predictive power and reliability of the model for the classification problem at hand.

4.1 Limitations

Despite the model's strong overall performance, several limitations were identified. The model demonstrated sensitivity to Photoshop altered genuine images, occasionally misclassifying them as fake due to artifact similarities with AI-generated content. Heavy retouching introduces visual artifacts such as texture irregularities and blending inconsistencies that closely resemble those found in AI-generated faces, causing the model to flag edited real images as synthetic. This indicates that the model's decision boundary is artifact-driven rather than semantically driven, making it susceptible to false positives in the presence of heavily post-processed images.

Furthermore, the model's improved performance on cropped facial images, while beneficial, reveals a dependence on region-specific features rather than scene-level understanding. This suggests that the model may under perform on uncropped or com-positionally diverse inputs where the facial region is not clearly iso-lated. Expanding the training dataset to include a broader range of edited genuine images and diverse facial compositions would likely reduce these sensitivities and improve the model's generalization.

Overall, enhancing the model's robustness against sophisticated image manipulations and reducing demographic biases are key steps to-ward improving its reliability and applicability in practical settings.

5. Conclusion

This research presents a DenseNet-121-based system that effectively detects AI-generated deepfakes, achieving 99.42% validation accuracy through transfer learning and fine-tuning with minimal computational demands. Its key contributions include a high-performing detection model, a user-friendly web application, and proven real-world utility, with only 65 false negatives and 11 false positives in 13,000 test images. The system supports digital authenticity in security, media, and identity verification, and its accessible interface encourages widespread adoption across industries.

The main contribution of this work lies in the integration of a high-performance DenseNet-121-based detection model with a user-oriented web application, enabling practical and accessible deepfake detection in real-world scenarios.

Acknowledgments

The authors are grateful to Er. Sushant Pandey, Project Supervisor, for his invaluable guidance, encouragement, and consistent support throughout the development of this paper. His expertise greatly contributed to the successful implementation of AI-generated face detection techniques. The authors would also like to thank the Com-puter and Electronics Department of Himalaya College of Engineering for providing the neces-sary resources and an environment conducive to research and collaboration. Appreciation is further extended to all team members for their collective effort, cooperation, and dedication in bringing this paper to fruition. The authors are grateful to Himalaya College of Engineering for this wonderful opportunity.

References

- [1] B. M. Gurusamy, P. K. Rangarajan, A. S., M. Sukesh, A. M., and J. Y., "Detecting ai-generated images with cnn and interpretation using explainable ai," 2024, pp. 1–6.
- [2] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [3] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.
- [4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, vol. 9905. Springer, 2016, pp. 21–37.
- [5] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2017.
- [6] M. K. Scheuerman, J. M. Paul, and J. R. Brubaker, "How computers see gender: An evaluation of gender classification in commercial facial analysis services," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, pp. 1–33, 2019.
- [7] Z. Guo, "Cartoon figure recognition with the deep residual network," in *2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*, 2021, pp. 157–160.
- [8] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5525–5533.
- [9] A. Khodabakhsh, R. Ramachandra, K. B. Raja, P. W. L. F., and C. Busch, "Fake face detection methods: Can they be generalized?" in *2018 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2018.
- [10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.
- [11] A. Borji, "Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and dall-e 2," *arXiv preprint arXiv:2210.00586*, 2022. [Online]. Available: <https://arxiv.org/abs/2210.00586>